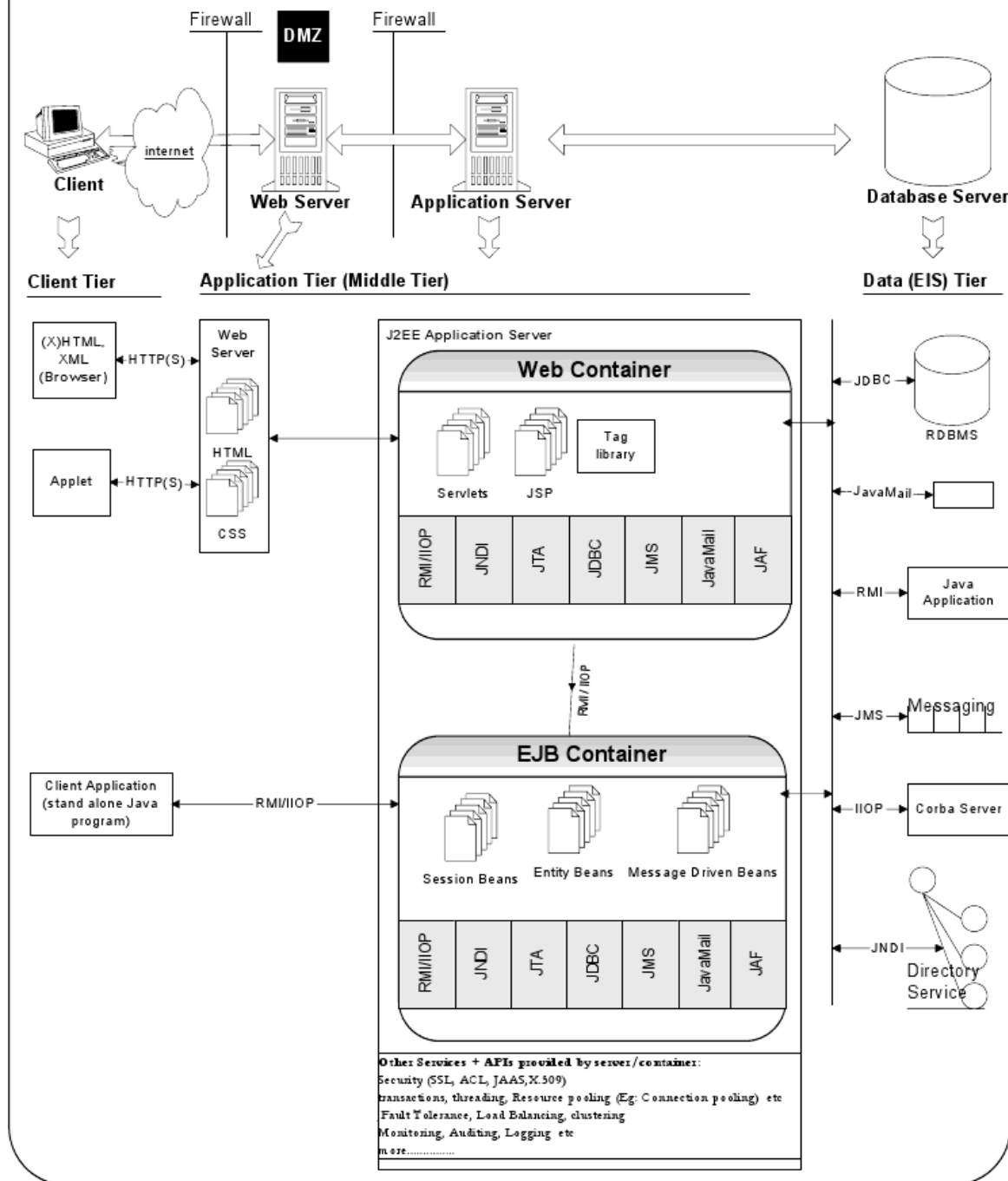


# Java Training 101

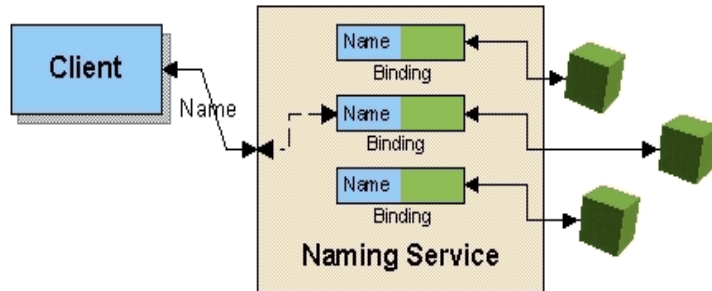
## Section 2

By  
Suraj

- JavaBeans - reusable software components for java
  - It is serializable (Ref: <http://java.sun.com/developer/technicalArticles/ALT/serialization>)
  - [Example] Create a JavaBean and use it in action
  - Create a class called PersonBean that implements `java.io.Serializable`
    - Add two attributes: "name", "ssn"
    - Create getters and setters for these variables
    - Create a constructor for this class (Ref: <http://download.oracle.com/javase/tutorial/java/javaOO/constructors.html>)
    - Create a tester for this class called TestPersonBean
    - In the main method, invoke the setter of the bean and print the value in a `System.out.println` command
- Generics - a programming construct that allows you to detect errors in your code before execution (during compilation)
  - Problem scenario - runtime passing of various types as "Object"
  - Solution - define the type in advance
    - Ex: methods - `Expense<Integer> exp = new Expense<Integer>();`
- EJB's - Enterprise JavaBeans architecture
  - Scalable, transactional, multi-user secure
  - Develop once - deploy in any JEE compliant application server
  - Provides system-level services, so focus is on developing business logic
  - Artifacts: Session beans, Entity beans, Message Driven beans
  - EJB 3 features - Annotations (ex: <http://www.fnogol.de/archives/2005/05/13/ejb-30-annotations-cheat-sheet>)



- JNDI - Java Naming and Directory Interface
  - A java API that allows you to easily interact with different file systems by making naming them. Ex. connect to a LDAP source or a file system on Unix or on Windows
  - How it works:
    - Get an initial context from a provider (ex. an LDAP service) and use it seamlessly



- JSP - Java Server Pages
  - Runs java code and html code in a common codebase
- Java applet - A stand-alone java program that runs inside a browser (Ex: <http://timeanddate.com/worldclock/personalapplet.html>)
- Containers - Web & EJB - the interface between a J2EE component and the low level platform specific functionality that supports J2EE components
- J2EE server - provider of system level support services, like security, transaction management, JNDI
  - *Leverages the J2EE architecture and provides configurable services*
  - *J2EE platform is multi-tiered*
  - *Client* - browser, applet      *Presentation* - servlet or JSP
  - *Business* - business logic      *Integration* - talk with external resources (like DB's)
  - *Resource* - database, Enterprise Resource Planning (ERP) systems

- J2EE platform
  - Components - ex: applets, JSP's
  - Services - ex: logging
  - API's
  - Protocols - ex: HTTP, TCP/IP, RMI, SOAP, SSL
- Web development frameworks
  - Why use? - Easy to use, ready-made widgets, allows abstraction of the interface from business logic (MVC and other design patterns) Ref: <http://www.javacamp.org/designPattern/mvc.html>
  - Top web development frameworks: <http://www.webdesignish.com/the-best-web-development-frameworks.html>
- SDLC - is the entire process of formal, logical steps taken to develop a software product (Ref: [http://en.wikipedia.org/wiki/Software\\_development\\_process](http://en.wikipedia.org/wiki/Software_development_process))

- [Hands-on Task] Review Java project setup in Eclipse
  - Ref: <http://www.vogella.de/articles/Eclipse/article.html>
- [Hands-on Task] Review Web project setup in Eclipse
  - Ref: <http://www.myeclipseide.com/documentation/quickstarts/webprojects>
- [Assignment] Review one Web development framework
  - Struts2 <http://java.dzone.com/articles/struts2-tutorial-part-27>
  - ICEfaces <http://wiki.icefaces.org/display/ICE/Creating+ICEfaces+Applications+with+Eclipse>

Acronym list <http://www.all-acronyms.com/tag/java>